Graph-based Image Segmentation Using Weighted Color Patch

Xiaofang Wang, Chao Zhu, Charles-Edmond Bichot, Simon Masnou

Graph-based Image Segmentation Using Weighted Color Patch

Xiaofang Wang[1], Chao Zhu[1], Charles-Edmond Bichot[1], and Simon Masnou[2]

[1] *Ecole Centrale de Lyon, LIRIS UMR5205, F-69134, France*
[2] *Université de Lyon, CNRS UMR 5208, Université Lyon 1, Institut Camille Jordan, F-69622 Villeurbanne, France*

Abstract

Constructing a discriminative affinity graph plays an essential role in graph-based image segmentation, and feature directly influences the discriminative power of the affinity graph. In this paper, we propose a new method based on the weighted color patch to compute the weight of edges in an affinity graph. The proposed method intends to incorporate both color and neighborhood information by representing pixels with color patches. Furthermore, we assign both local and global weights adaptively for each pixel in a patch in order to alleviate the over-smooth effect of using patches. The normalized cut (NCut) algorithm is then applied on the resulting affinity graph to find partitions. We evaluate the proposed method on the Prague color texture image benchmark and the Berkeley image segmentation database. The extensive experiments show that our method is competitive compared to the other standard methods with multiple evaluation metrics.

Keywords: Image segmentation, weighted color patch, affinity graph, normalized cuts.

1 Introduction

Image segmentation is one of the fundamental yet most difficult tasks in computer vision. In recent years, the graph-based methods have been proven successful and widely applied to image segmentation, mainly because they have an efficient tool to solve the optimization problem of segmentation [2] and can naturally incorporate different type of features in the affinity graph. Usually, the graph-based methods first construct an affinity graph from a given image, and then partition the resulting graph into different clusters with certain cut criterions [10] [19]. Thus, constructing a discriminative affinity graph plays an essential role in such methods. For a desirable partition result, the pixels should be similar to each other in intra-clusters while different from each other in inter-clusters. The similarity between two pixels can be measured by the distance of different features such as color, boundary, texture, etc.

Therefore, feature is an important factor since its properties directly influence the discriminative power of the resulting affinity graph. In the literature, numerous works have been proposed to design powerful features for image segmentation. Generally, the features applied to the graph construction can be categorized as pixel-based and region-based according to the definition of graph nodes. For the pixel-based features, pixels in an image are directly considered as nodes in the affinity graph. Brightness, color and boundary are the most common features adopted to compute the pairwise similarity of pixels. Moreover, inventing contour cue [13] has been proposed to capture edge information. Color and contour cue [7] are often combined in an unsupervised manner, and some works tried to fuse multiple types of features by learning on the ground-truth data of image database [4]. In the region-based case, graph nodes are represented by the over-segmented superpixels [14] [15]. Many kinds of features can be explored to compute the similarity of two nodes. For example, [15] used the averaged color to compute the affinity graph, and [5] fused color histogram, local binary patterns (LBP) and scale invariant feature transform (SIFT) with low-ranking.

In this paper, we propose a new method based on the weighted color patch to construct a more discriminative affinity graph. The idea of representing a pixel with a patch has been proven successful in non-local image denoising [3]. However, it produces the over-smooth effect due to considering each member equally in the patch. Therefore, it is necessary to assign different weight to each pixel in the patch. J. Zexuan et al. [12] investigated this idea in their work on fuzzy c-means clustering, but they only considered gray intensities to compute the similarity of two pixels. For image segmentation, it is insufficient to use only gray intensities, while color is also a very discriminative and efficient feature for identifying different objects, especially in natural images. Therefore, our proposed method intends to incorporate both color and neighborhood information. There are two main advantages: i) it can smooth local regions by averaging color information and ii) it can capture texture information by considering context neighboring cue. Furthermore, in order to incorporate spatial information, we also propose to assign a global weight to each pixel in an image according to different proportion of the object and background, so that the contrast between them is enhanced and a more discriminative affinity graph is constructed.

The rest of the paper is organized as follows: we introduce the proposed weighted color patch (WCP) method elaborately in section 2, where local and global weights are presented in section 2.1 and 2.2 respectively, and we introduce the affinity graph construction based on WCP in section 2.3; in section 3, we present extensive experiments on the Prague texture image benchmark [11] and the Berkeley image segmentation database [1], and report the quantitative results with associated multiple evaluation metrics; the conclusions are drawn in section 4.

2 Proposed method for image segmentation

In this section, we present the proposed weighted color patch (WCP) method in detail and introduce the affinity graph construction based on WCP for image segmentation. The basic idea of WCP is to represent a pixel with a patch around it and assign both local and global weights to each member in the patch. To incorporate color information, the weighted patch is first calculated in each channel of the RGB color space, and then combined in the affinity graph construction step. For image segmentation, we apply the proposed method to compute the weight of edges in the affinity graph, which is further partitioned by the normalized cut (NCut) algorithm [19].

2.1 Local weights computation

As introduced in the introduction, using patches directly will cause the over-smooth effect mainly due to considering each member in the patch equally. Therefore, it is necessary to assign different weights to different pixels. In this paper, we adopt the method described in [12] to compute the local weights adaptively.

Let an image represented by $I = \{g_1, ..., g_x\}$ with g_x as pixel intensity, and a patch vector denoted as $P_k = (g_k, N_k)$, where N_k is the neighborhood around the central pixel g_k with the size $w \times w$. For each pixel g_r in the patch, its mean-square deviation σ_r is defined as follows:

$$\sigma_r = \Big[\frac{\sum_{n \in N_k \setminus \{r\}} (g_r - g_n)^2}{n_k - 1}\Big]^{1/2} \tag{1}$$

The computed mean-square deviation σ_r is then applied in the following exponential kernel function:

$$\xi_r = exp\Big[-\Big(\sigma_r - \frac{\sum_{r \in N_k} \sigma_r}{n_k}\Big)\Big] \tag{2}$$

Finally, the local weight of pixel g_r is obtained by normalizing the value of ξ_r:

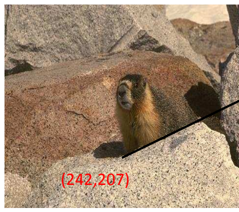
$$\omega_r = \frac{\xi_r}{\sum_{r \in N_k} \xi_r} \tag{3}$$

Since the applied Gaussian kernel decays very fast, those pixels whose mean-square deviation is far away from the average value will have a relatively small weights. An illustration of how to calculate the local weights is shown in Fig.1, and we take a patch from a natural image to depict the effectiveness of the local weights. We can observe that the patch is extracted from an inhomogeneous boundary region, thus relative to the central pixel, those pixels lying on the other side of the boundary are assigned with smaller weights in order to decrease their impact to the patch.

2.2 Global weights assignment

In addition to the local weights, which only reflect the structure of a local patch, we also propose to assign a global weight to each pixel in an image according to different proportion of the object and background, since we observed that they should have different contribution to the affinity graph construction because of different structure of the whole image content. More precisely, the proposed global weights are obtained by calculating a normalized histogram of the image based on the pixel values.

Fig.2 presents an example to show the effectiveness of using the global weights. Suppose that the intensities of the background, the triangle object



44	44	43
57	48	59
128	102	127

Gray values

47.0638	47.0638	47.7507
39.6626	44.4635	38.8346
69.1158	47.8278	68.2138

σ

0.0002	0.0002	0.0001
0.3031	0.0025	0.6938
0.0000	0.0001	0.0000

ω

Figure 1: An illustration of the local weights calculation of a patch extracted from the boundary region in a natural image (the first column shows the gray values, the second column is the mean-square deviation of each pixel, and the last column shows the weights assigned to each pixel).

Figure 2: An illustration of the effectiveness of the global weights in a synthetic image.

and the rectangle object are 255, 200, 100 respectively, then their calculated global weights will be 0.7, 0.1 and 0.2 respectively. Without the global weights, the distances between the background and the objects are 55 and 155 respectively, while both distances become 158.5 when considering the global weights. Thus we can see that i) the distances between the background and the objects are increased; ii) both objects have the same distance to the background, which makes them easier to be segmented simultaneously.

2.3 Affinity graph construction

Given an image I, it can be represented as a graph $G = (V, E)$, with V being the set of vertices and E being the set of edges connecting two vertices. We apply the proposed WCP method to compute the weight of edges in the graph. In order to incorporate color information, the affinity graph is first computed in each channel of the RGB color space, formally defined as follows:

$$W(x_i, x_j) = e^{-(\|\sum_{i=1}^{N} P^c_{wi} - \sum_{j=1}^{N} P^c_{wj}\|^2/\sigma)}, \; ||x_i - x_j||_2 < r \tag{4}$$

where W is the affinity graph, and $W(i, j)$ defines the edge weight of two vertices i and j in the graph. According to the derived weights, we discard those pixels in the patch whose weights are smaller than a threshold value which is set to $1/(n_k) \times 1/N$ with n_k the size of the local patch, and N the total number of pixels in the image. x_i represents the spatial coordinates of pixel i, and r is the graph radius.

$$P_{wi} = (g_r,\ r \in N_k,\ \ if\ \ \omega_r \times \xi_r >= (1/(n_k) \times 1/N)) \tag{5}$$

with ξ_r represents the global weight assigned to pixel g_r. σ in Eq.(4) is a positive constants to control the decaying speed of gaussian kernel function. c represents each channel of the RGB color space.

The final affinity graph is obtained by averaging the results from all the channels.

2.4 Graph partitioning

Given the affinity graph W, we apply the normalized cut (NCut) algorithm to partition the graph into k groups by solving the following generalized eigen-vector problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{6}$$

where $L = D - W$ is the Laplacian matrix, $D = diag(W\mathbf{1})$ is the diagonal degree matrix. The bottom k eigenvectors are computed either by k-means [20] or discretization method [19].

3 Experimental Evaluation

In this section, we evaluate the proposed WCP method for image segmentation on two popular databases: the Prague color texture benchmark [11] and the Berkeley image segmentation database (BSD) [1]. For simplicity, we fix the parameters for all the following experiments as: $\sigma = 10$, $r = 10$ in Eq.(4) and the patch size is 7×7.

3.1 Results on Prague texture benchmark

The Prague texture benchmark datasets are computer generated 512×512 random mosaics filled with randomly selected textures. This benchmark provides a bunch of criterions for evaluation (see Table 1), and we refer the readers to the website of [11] for a detailed description of all the measurements. The proposed method is compared with the other unsupervised benchmark algorithms, including: EDISON [6], JSEG [23] and SWA [18]. Fig. 3 presents seven selected 512×512 experimental benchmark mosaics and Table 1 gives their corresponding numerical scores w.r.t. different indicators. It can be observed that EDISON and JSEG tend to oversegment images while SWA and our method have better trade-off between over-/under-segmentation. From the results presented in Table 1, we can see that no single algorithm can outperform all the others on all the measurements. However, our method ranks the first place on 7 indicators (displayed in bold) while JSED has only two and SWA has only four best results. In particular, although EDISON also has 8 best performances, its other performances such as OS, O and C lagged far behind ours, which makes our method the best overall algorithm regarding to all associated indicators.

Metrics	region-based					consistency measure		clustering			-
Methods	CS↑	OS↓	US↓	ME↓	NE↓	GCE↓	LCE↓	dM↓	dD↓	dVI↓	-
EDSION	12.68	86.91	**0.00**	**2.48**	**4.68**	**3.55**	**3.44**	35.37	16.84	25.65	-
JSEG	27.47	38.62	5.04	35.00	35.50	18.45	11.64	23.38	15.19	17.37	-
SWA	27.06	50.21	4.53	25.76	27.50	17.27	11.49	24.20	**13.68**	17.16	-
WCP	**30.92**	**4.12**	26.67	37.40	35.72	20.28	14.82	**22.27**	16.83	**13.25**	-
Metrics	pixel-wise										
Methods	O ↓	C↓	CA↑	CO↑	CC↑	I.↓	II.↓	EA↑	MS↑	RM↓	CI↑
EDSION	73.17	100.00	31.19	31.55	**98.09**	68.45	**0.24**	41.29	31.13	**3.21**	50.29
JSEG	37.94	92.77	**55.29**	61.81	87.70	38.19	3.66	66.74	**55.14**	4.96	70.27
SWA	**33.01**	85.19	54.84	60.67	88.17	39.33	2.11	**66.94**	53.71	6.11	**70.32**
WCP	41.32	**28.70**	53.55	**67.49**	63.39	**32.51**	6.60	62.69	51.23	9.34	64.00

Table 1: Quantitative comparison of our results with other methods on the Prague benchmark with multiple measurements.

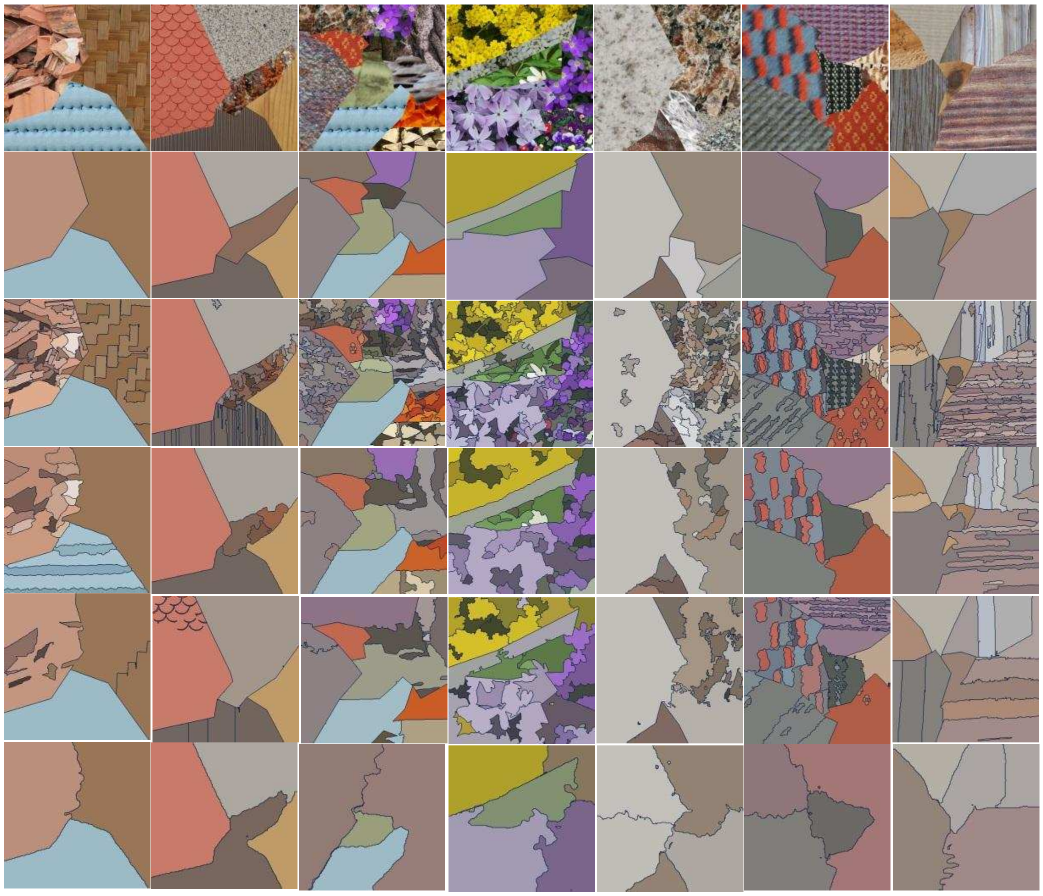


Figure 3: Visual comparison of our results with other methods on the Prague benchmark (examples presented in row-wise, from up to down, are respectively the original images, ground truth maps, EDISON, JSEG, SWA and our results).

3.2 Results on Berkeley image database

The Berkeley image database contains 300 images and their corresponding ground truth (each image has at least 4 human annotations). In our experiments, we test the proposed method on all the 300 images, since the algorithm has no parameter to be trained. The number of segments k is set from $[3, 5, 7, 10, 12, 15, 18, 20, 23, 25, 28, 30, 31, 32, 35, 40]$. The final results are evaluated according to 4 associated measurements, including: Probabilistic Rand Index (PRI) [21], Variation of Information (VoI) [17], Global Consis-

Figure 4: Visual comparison of our results with other methods on the Berkeley database (examples presented in column-wise, from left to right, are respectively the original images, NCut, GBIS and our results).

tency Error (GCE) [16], and Boundary Displacement Error (BDE) [9]. The popular NCut, GBIS[8] and Normalized Tree Partitioning (NTP) [22] are applied for the purpose of comparison, and their parameters are the same as [14], which manually tuned the number of segments for each image.

The quantitative results are presented in Table 2, with the best results highlighted in bold for each measurement. It is obvious to see that the proposed WCP method ranks the first place with respect to VoI and BDE compared with the other methods. Fig.4 presents some visual comparisons of our results with the other methods, and we can see that NCut tends to split homogenous large region into separate regions and GBIS has thick edges, while our proposed method can obtain more meaningful region with accurate boundary. We also present some examples segmented by our proposed method in Fig.5. It can be observed that our method can well segment the texture images (the penguin, the leopard, web girl), and it has high discriminative power to detect objects from different backgrounds.

Figure 5: Some examples segmented by our method on the Berkeley database.

Methods	PRI↑	VoI↓	GCE↓	BDE↓
NCut	0.7242	2.9061	0.2232	17.15
GBIS	0.7139	3.3949	**0.1746**	16.67
NTP	**0.7521**	2.4954	0.2373	16.30
WCP	0.7496	**2.4399**	0.2392	**15.7416**

Table 2: Quantitative comparison of our results with other methods on the Berkeley database with multiple measurements: the results of our method are obtained over the best tuned parameter for each image.

4 Conclusion

In this paper, we propose a new method based on the weighted color patch to construct the affinity graph for image segmentation. The proposed method is invariant to uneven light conditions and noise benefitting from the usage of image patches. Furthermore, we assign a local weight to each member in the patch to overcome the over-smooth effect, and also calculate a global weight for each pixel in the image to enhance the contrast between the background and the objects. The proposed method is evaluated by extensive experiments on two popular segmentation databases, and is quantitatively compared with some other standard algorithms. The results show that our method is powerful and competitive, and can be further applied on other clustering problems.

Acknowledgements This work was supported by the Chinese Scholarship Council (CSC), and by the French research agency ANR through the VideoSense project under the grant 2009 CORD 026 02.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[2] P. Bo, Z. Lei, and Z. David. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020–1038, 2013.

[3] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. In *CVPR*, pages 60–65, 2005.

[4] J. Malik C. Fowlkes, D. R. Martin. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *CVPR*, pages 54–64, 2003.

[5] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, pages 2439–2446, 2011.

[6] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *ICPR*, pages 150–155, 2002.

[7] T. Cour, F. Bénézit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, pages 1124–1131, 2005.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[9] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV*, pages 408–422, 2002.

[10] L. W. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[11] M. Haindl and S. Mikes. Texture segmentation benchmark. In *ICPR*, pages 1–4, 2008.

[12] Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia, and D. D. Feng. Fuzzy c-means clustering with weighted image patch for image segmentation. *Appl. Soft Comput.*, 12(6):1659–1667, 2012.

[13] M. Jitendra, B. Serge, L. K. Thomas, and S. Jianbo. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, 2001.

[14] T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. In *CVPR*, pages 2101–2108, 2010.

[15] Z. Li, X. Wu, and S. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *CVPR*, pages 789–796, 2012.

[16] D. R. Martin, C. Fowlkes, D. Tal, , and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[17] M. Meila. Comparing clusterings: an axiomatic view. In *ICML*, pages 577–584, 2005.

[18] E. Sharon, A. Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. In *CVPR*, pages 469–476, 2001.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[20] V. L. Ulrike. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[21] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *PAMI*, 29(6):929–944, 2007.

[22] J. Wang, Y. Jia, X. Hua, C. Zhang, and L. Quan. Normalized tree partitioning for image segmentation. In *CVPR*, 2008.

[23] D. Yining and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *PAMI*, 23(8):800–810, 2001.